

ORSZÁGOS KÖZÉPTÁVÚ KUTATÁSI-FEJLESZTÉSI TERV



AZ ATOMERŐMŰVEK BIZTONSÁGOS ÜZEMELTETÉSÉT
SZOLGÁLÓ KUTATÁSI-FEJLESZTÉSI FELADATOK

PK-011

2.8.1

A VVER-440 TÍPUSÚ ATOMERŐMŰVEK
NEUTRONFIZIKAI ÜZEMVITELI ADATAINAK
A RENDSZERE, AZ RFIT-ED PROGRAM

KFKI

1981.

KFKI-1982-70

SZATMÁRY Z.

A VVER-440 TÍPUSÚ ATOMERŐMŰVEK
NEUTRONFIZIKAI ÜZEMVITELI ADATAINAK A
RENDSZERE. AZ RFIT-ED PROGRAM

Hungarian Academy of Sciences

CENTRAL
RESEARCH
INSTITUTE FOR
PHYSICS

BUDAPEST

A VVER-440 TÍPUSÚ ATOMERŐMŰVEK
NEUTRONFIZIKAI ÜZEMVITELI ADATAINAK A RENDSZERE
AZ RFIT-ED PROGRAM

Szatmáry Zoltán
Központi Fizikai Kutató Intézet
1525 Budapest 114, Pf.49

HU ISSN 0368 5330
ISBN 963 371 956 9

KIVONAT

A program üzemelő atomerőművekben mért adatok rendszerezett tárolására szolgál. Az adatok részenként vagy egészben előhívhatók újabb információ beírására, javítására vagy további kiértékelés céljára. Az adattárolás subfile-okban történik, az információ részben szöveges formátumú, részben numerikus. A program egyszerű saját utasításkészlettel rendelkezik, amelyekkel az összes szolgáltatása vezérelhető.

АННОТАЦИЯ

Программа предназначена для систематического сохранения эксплуатационных данных энергетических ядерных реакторов. Данные могут быть представлены полностью или частично для дальнейших исправлений, добавлений или обработки. Сохранение данных происходит в подфайлах (subfile), часть информации сохраняется в виде обычного текста, остальные в нумерической форме. Программа снабжена собственными управляющими командами, с помощью которых выполняются все задачи.

ABSTRACT

The program is used for the systematic storage of operational measurement information from power reactors. The data can be accessed in full or any part of it according to operator request, for corrections, adding new information, or later evaluation. Data storage is arranged in subfiles, part of the information is in text form, the rest is numerical. The program has its own simple set of instructions which serve to initiate the various functions and options.

I. BEVEZETÉS

Energetikai reaktorok üzemeltetéséhez egy kísérletileg ellenőrzött, esetleg azokhoz illesztett számítógépi modellre van szükség. Ezt a programot használjuk az átrakások tervezésére, az üzemvitel optimalizálására, az üzemnyomonkövetésére, bizonyos rendkívüli üzemállapotok analizisére. A rendelkezésre álló modell a BIPR program [1]. Ellenőrzése és bizonyos konstansainak az illesztése a novovoronyezsi reaktorblokkok üzemviteli adatainak a felhasználásával történt. Ezt a modellt használja minden ország, amely VVER-440 típusu atomerőműveket üzemeltet. Egyik sem tekinti azonban a modellnek a jelenlegi állapotát lezártnak, mert általános az a vélemény, hogy a számítási pontosság fokozható. A modell pontossága közvetlenül kihat a reaktor maximálisan megengedhető teljesítményére. Az elfogadott megítélés szerint az ezzel a programmal számítható teljesítményeloszlás pontossága 10%-nál nem jobb, ezért a radiális egyenlőtlenségi tényező számított értékére vonatkozóan egy 1,10 értékű biztonsági tényező használata szükséges [2]. Egyéb számított mennyiségek, mint pl. a szabályozó rudak értékessége, a bórsav hatása a reaktivitásra, a kampány hossza számítási pontosság szempontjából igazában nincs még kellő részletességgel és alapossággal analizálva.

Az Ideiglenes Nemzetközi Kollektiva keretén belül merült fel az a gondolat, hogy a számítási modell pontosságának a vizsgálatára és pontatlanságainak a kiküszöbölésére kiválóan alkalmasak lehetnek a működő VVER-típusú energetikai reaktorblokkok neutronfizikai üzemviteli adatai, ha azok egy ilyen analízis által megkívánt rendszerességgel és teljességgel rendelkezésre állnak. A teljességre két tekintetben is szükség van. Egyrészt minél több reaktorra kell ilyen adatokat gyűjteni, másrészt az egy reaktorra vonatkozó adatok gyűjtését a mért mennyiségek minél szélesebb körére ki kell terjeszteni. Így lehet csak ugyanis a technológiai bizonytalanságokat kiszűrni és általánosítható következtetésekre jutni. Egy energetikai reaktoron mérhető mennyiségekhez sohasem lehet olyan pontosan meghatározni a mérés körülményeit, mint egy kritikus rendszer esetében, így az energetikai reaktorokon végzett mérések inherensen sokkal pontatlanabbak, mint a kritikus rendszereken

végzettek. Ezt csak a mérések nagy száma és a mérési eredmények rendszeres-
sége ellensúlyozhatja. Ezért beszélünk az üzemviteli adatok rendszeréről.

Egy ilyen adatrendszer létrehozásához három lényeges dolog kell: (1) annak eldöntése, milyen adatok kerüljenek be a rendszerbe, (2) a rendszer kiszolgálására szolgáló program és (3) a rendszer használatát biztosító mód-
szerek és programok. Ezek közül az első kettő elengedhetetlen ahhoz, hogy az adatok gyűjtése és rendszerbe való felvétele megkezdődhessen. Az eddigi-
ekben ez a két dolog tisztázódott. Az Ideiglenes Nemezetközi Kollektíván be-
lül dolgoztuk ki az üzemviteli adatoknak azt a rendszerét, amelynek a leírá-
sát a jelen dolgozatban adjuk meg. Ebben a munkában figyelembe vettük a VVER-
440 típusu atomerőműveket üzemeltető országok (Szovjetunió, NDK, Bulgária)
szakembereinek a véleményét. Ily módon a rendszer alkalmas arra, hogy az e-
zekből az országokból származó üzemviteli adatokat tárolja. Az alábbiakból
látni fogjuk, hogy a Paksi Atomerőmű igényeit is ki tudja elégíteni.

A rendszer kiszolgálására szolgál a RFIT-ED program. Ez a ZR-6 kritikus
rendszeren végzett mérések hasonló rendeltetésű rendszerének a kezelésére és
használatára kidolgozott RFIT program tapasztalatainak a felhasználásával
született. Maga a RFIT-ED név azt kívánja hangsúlyozni, hogy a két adatrend-
szer szelleme és elvi alapjai azonosak. Természetesen jelentős tartalmi kü-
lönbségek vannak a két rendszer között, mert a ZR-6 adatrendszer primér, te-
hát kiértékeletlen kísérleti információt tartalmaz, az üzemviteli adatok
rendszere viszont már kiértékelte mérési eredményeket. Ebből a két adatrend-
szer és a két program között jelentős különbségek adódtak.

A javaslatunk az, hogy a Paksi Atomerőmű reaktorblokkjain mért adato-
kat kezdettől fogva az alább ismertetett rendszerben gyűjtsük. A fentebb em-
litett országok üzemviteli adataikat készek egy közös adatkönyvtár számára
rendelkezésre bocsátani. Az adatok átadásával kapcsolatban azonban felmerült
egy gyakorlati probléma. Mindegyik országban létezik egy helyi adatrendszer,
amely tartalmilag az RFIT-ED rendszerrel lényegében azonos információt tar-
talmaz, de az adatok átadásához még szükséges a helyi rendszerek és az RFIT-
ED rendszer közötti "fordító programokat" megírni. Így válna az RFIT-ED rend-
szer a különböző helyi rendszerek közötti interface-szé. (Az, hogy valamelyik
helyi rendszer váljon ilyen központi rendszerre, fel sem merült, mert a helyi
rendszerek tulságosan partikulárisak.) Ez a körülmény bizonyos mértékig pri-
vilegizált helyzetbe hozhatja a Paksi Atomerőművet, mert az ott létező helyi
rendszer azonos lehet a közös rendszerrel. Az említett fordító programok még
nem születtek meg, de hosszabb távon majd beépülnek az RFIT-ED programba, a-
honnán mint speciális opciók hívhatók lesznek.

Az energetikai reaktorok rendeltetése az energiatermelés. Ennek a tri-
viális megállapításnak a kijelentésére azért van szükség, hogy megértsük a
létrejött rendszer felfogását. Ez a rendszer ugyanis csak olyan adatokat tar-
talmaz, amelyek a normális üzemmenet során mindenképpen keletkeznek, tehát a
rendszernek adatokkal való feltöltése nem igényel járulékos méréseket. Amire

szükség van, az csupán az üzemi mérések gondos elvégzése és kiértékelése, a kapott eredmények közül a reprezentatív adatok kiválasztása, és ezeknek a megadott szabályok szerint való rendezése.

Az alábbiakban először ismertetjük az üzemviteli adatok rendszerét, majd az RFIT-ED program használatának az általános szabályait. Ebben az utóbbi leírásban a [3] riportra támaszkodunk.

II. A NEUTRONFIZIKAI ÜZEMVITELI ADATOK RENDSZERE

Az adatrendszer ismertetését két részre bontjuk: először ismertetjük, milyen üzemállapotokra vonatkozóan tartjuk célszerűnek az adatok gyűjtését, majd rátérünk magukra a tárolandó mennyiségekre. Mint a III. részben látni fogjuk, a rendszer szerkezete nagyon flexibilis, tehát a most elmondandókhoz képest könnyen továbbfejleszthető, és a felhasználók belátásától függően elég sokféleképpen használható.

II.1. A TEKINTETT ÜZEMÁLLAPOTOK

A következő üzemállapotokra vonatkozóan javasoljuk elsősorban az üzemviteli adatok tárolását:

A) A reaktor üzeme állandó teljesítményen

A reaktor az üzemidő legnagyobb részében állandó vagy közel állandó teljesítményen üzemel. Az ilyen állapotokat legegyszerűbben úgy határozhatjuk meg, hogy kiegészítő számításokkal való nyomonkövetésük nem igényli tranziens folyamatok figyelembevételét (tehát fel lehet például tételezni, hogy a Xe egyensúlyi koncentrációban van). A teljesítmény napnál hosszabb időskálán lassan változhat, kismértékű gyors teljesítményváltozások pedig jó közelítéssel figyelembe vehetők az effektív napokon keresztül.

B) A reaktor üzeme változó teljesítményen

Vannak a reaktornak olyan üzemállapotai, amikor a tranziens folyamatok lényeges szerepet játszhatnak, tehát amikor a teljesítmény e folyamatok időskáláján belül változik. Ezek már csak rossz közelítéssel vehetők figyelembe az effektív napok segítségével, ha magát a változást akarjuk analizálni. Ilyen állapotok lehetnek:

- a teljesítmény változása meghatározott napi program szerint (ilyenek főleg a kampány végén szoktak előfordulni),
- Xe-lengések,
- teljesítménytényezőn való üzemelés a kampány végén, stb.

Arról van tehát szó, hogy egy adott kampány globális lefutása megérthető és végigszámolható, ha ilyen változásokat csak az effektív napok számolásában

veszünk figyelembe, de a teljesítményt lassan változónak ábrázoljuk, viszont bizonyos tudományosan érdekes részletek bemutatása érdekében néhány rövidebb időszak változásait változó teljesítményű állapotok formájában is regisztráljuk.

c) Indítási kísérlet

A fizikai és energetikai indítás során számos olyan mérést végeznek el, amelyekre a fenti két üzemállapotban nincs lehetőség, viszont eredményeik nagyon fontosak a későbbi üzemmenet megértése és tervezése szempontjából.

* * * * *

Az adattároló rendszer jelenlegi formájában ezeknek az üzemállapotoknak megfelelő adatok fogadására alkalmas. Elképzelhetők természetesen további üzemállapotok is. Ilyenek bizonyos üzemzavari helyzetek, amelyek lefolyását érdemes lehet regisztrálni annak reményében, hogy egy későbbi analízis hasznos tanulságokkal szolgálhat. Nem sok publikáció szól például arról, hogyan reagál a reaktor egy leszakadt szabályozó rudra, hogyan alakul ennek hatására a teljesítményeloszlás, mennyi ideig érezteti ez a hatását stb. Nem valószínű, hogy ilyen adatok valaha is nyilvánosságra kerülnek, de egy illetéktelenek előtt jól védett adatrendszerben minden nehézség nélkül rögzíthetők. Egy további példa lehet az u.n. perturbációs mérés által szolgáltatott adatsor. Tekintve, hogy ilyen adatok sem állnak még rendelkezésre, nehéz lenne ma még megmondani, alkalmas-e tárolásukra a jelenlegi rendszer. Ha nem alkalmas, valószínűleg kis módosításokkal alkalmassá tehető erre. Amikor azt mondtuk, hogy a rendszer flexibilis, ezen elsősorban azt értettük, hogy ilyen módosításokra bármikor lehetőség legyen a korábban már megszerkesztett adatmező változtatása nélkül.

11.2. A TÁROLANDÓ MENNYISÉGEK

A különböző üzemállapotokban keletkező üzemviteli adatok elsősorban kritikus állapotok adatai. Anélkül, hogy ismernénk azoknak a kritikus állapotoknak a jellemzőit, amelyek a kampány során megvalósultak, nem lehet sem a kampányt végigszámolni, sem pedig a különböző mérési eredményeket értelmezni, vagy általában a reaktor viselkedését megérteni. Más szavakkal ez azt jelenti, hogy az üzem során megvalósult kritikus állapotok sorozata minden analízis és számítás nélkülözhetetlen bemenő adata. Amikor egy kísérleti eredményt tekintünk, akkor a kísérlet elvégzésének az időpontjáig megvalósult kritikus állapotok sorozatát mint a reaktor előtörténetét szoktuk emlegetni. Ebből következik az adattároló rendszernek az a szemlélete, hogy alapvetőnek tekinti a kritikus állapotok sorozatára vonatkozó adatsort, és feltételezi, hogy a

többi mérési adat ennek a sorozatnak valamelyik tagjához van rendelve. Tehát bármilyen más adattól el lehet esetleg tekinteni, de a kritikus állapotok sorától nem.

A kritikus állapotok jellemzésére szolgál mindenekelőtt a kartogramma, továbbá olyan mennyiségek értéke, mint rudhelyzetek, borkoncentráció, hőmérséklet, az effektív napok száma stb. Ez utóbbi mennyiségek az üzem során állandóan változtatják értéküket, így felmerül a kérdés, milyen gyakorisággal célszerű ezeket az értékeket a rendszerben rögzíteni. Éppen azért definiáltuk az előbb a különböző üzemállapotokat, mert ezek elsősorban a változások időskálájában különböznek egymástól. Például, állandó teljesítményen való üzemállapotban elégséges minden naptári napra vonatkozóan (vagy ritkábban) egy jellemző vagy átlagos kritikus állapot jellemzőit feljegyezni. Tranziens folyamat esetében a feljegyzés gyakoriságának illeszkednie kell magához a folyamathoz. Így például, egy Xe-lengés lefolyásához a kritikus állapotokat néhány órás gyakorisággal kell feljegyezni. Indítási kísérletben az idő alig játszik szerepet, hiszen itt kiégés még nincs, úgyhogy elégséges a jól definiáltan megvalósított kritikus állapotokat időadat nélkül rögzíteni. Mindezek természetesen csak ajánlások, az adatok megadásának a módja és gyakorisága mindenképpen az adatok összeállítójának a szempontjaitól függ. A rendszer kellően általános ahhoz, hogy eléggé szerteágazó szempontok figyelembevételét megengedje. A továbbiakban a kritikus állapotokra vonatkozó mennyiségeket "alapmennyiségeknek" fogjuk nevezni. Hogy ezek pontosan mik, azt a következő részben adjuk meg részletesen.

Az atomerőműben további mérési adatok is keletkeznek: szabályozó rudak értékségei (differenciális és integrális), bórsavértékség, hőmérsékleteloszlás, kazettánkénti kiégés stb. Ezeket a mennyiségeket ritkábban határozzák meg, mint az alapmennyiségeket, sőt vannak reaktorblokkok, ahol nem is áll mindegyik rendszeresen rendelkezésre. Jóllehet nagyon hasznosak és részletes analízis számára nélkülözhetetlenek, formálisan mégsem feltételei annak, hogy a kampányt számítással nyomonkövessük. Ezért a rendszer ezeket kiegészítő információknak tekinti. Ez azt jelenti, hogy az RFIT-ED program az alapmennyiségek megadása nélkül nem fogadja el a többi mennyiséget, de az alapmennyiségeket ez utóbbiak nélkül is elfogadja.

Mint fentebb, az üzemállapotok jellemzésénél már utaltunk rá, egy reaktorblokk adott kampányáról akkor rendelkezünk igazán érdekes és tanulságos adatokkal, ha a kampány egészének a történetét egy-két napos időközönként regisztrált kritikus állapotok sorozatával (mint állandó teljesítményű állapotokéval) globálisan leírjuk, és ezt kiegészítjük

- bizonyos feljegyzett állapotokban végzett hőmérsékleteloszlás, rudértékség stb. mérések eredményeivel,
- bizonyos részletesen regisztrált tranziens folyamatok kritikusállapotadataival, esetleg egyéb adataival (természetesen pontosan megjelölve a folyamat kezdetének az időpontját).

Különös figyelmet érdemelnek a kampány végének az adatai. Tekintve, hogy minden átrakás az előző kampány végén kapott összetétel alapján számított séma szerint történik, a számítás pontosságát erősen befolyásolhatja, milyen pontosan ismerjük, hogyan állt le a reaktor: a teljesítmény hirtelen csökkent nullára, a teljesítménytényezőn járt egy darabig, vagy más módon. Szórványos adatokat talált a szerző arra vonatkozóan, hogy a kampány vége felé a rudértékességek, a hőfoktényező és több más ilyen mennyiség értéke nagyon eltérhet attól, amit az indítási kísérlet során mértek, tehát megadása a számítások pontosítása, de még a teljesítménytényezőn való üzem gyakorlati kérdései szempontjából is nagyon fontos lehet.

Szükséges néhány szóval indokolni, miért hiányoznak a tárolandó mennyiségek közül a termohidraulikai vonatkozású adatok. Teljesen logikus és jogos az az ellenvetés, hogy olyan adatok, mint a melegjáratás adatai, a szekundérköri hőmérleg és hasonló legalább annyira érdekesek és fontosak, mint a neutronfizikai adatok. Hogy ezek kimaradtak, annak praktikus okai vannak. Tömegük sokkal nagyobb a neutronfizikaiakénál. Így nehezebben is rendszerezhetőek. Ezért általánosan az volt a vélemény, hogy tulságosan sokat akarnánk egyszerre markolni, ha úgy indulnánk neki az adatok rendszerezésének, hogy mindent rögtön az első menetben össze akarnánk gyűjteni. Ha olyan döntés születik, hogy ez utóbbiak rendszerezése is szükséges, a neutronfizikai adatok rendszere egy első lépésnek tekinthető egy teljesebb adatrendszer kiépítésének az útján. Így a jelenlegi rendszer termohidraulikai és termodinamikai adatokat csak olyan mértékben tartalmaz, amilyen mértékben a neutronfizikai adatok felhasználásához szükséges.

Általános tapasztalat, hogy a számszerűen megadott adatmezőket hasznos magyarázó kommentárokkal kiegészíteni. Ezek vonatkozhatnak olyan triviális dolgokra, mint az adatok forrása, az adatok összeállítójának a neve, esetleges irodalmi hivatkozások stb, de a leghasznosabbak azok a kommentárok, amelyek a közölt adatok felhasználását illetően igazítanak utba. A már említett ZR-6 adatrendszerrel kapcsolatban tapasztaltuk, hogy a kísérletezők lebecsülik e kommentárok fontosságát. Ez tulajdonképpen érthető. A frissen mért adatok összeállításakor még minden világos, mindenre emlékezünk, és azt gondoljuk, hogy ez néhány év múlva is így lesz. Szükségtelen többletmunkának tűnik leírni, milyen korrekciókat alkalmaztunk az egyes mért adatok kiértékelése közben, melyek voltak azok az adatok, amelyek helyességében magunk sem voltunk egészen biztosak, de a teljesség kedvéért mégis úgy döntöttünk, hogy megadjuk őket, mi volt a célunk egy adatsor összeállításával, hogyan becsültük a hibákat stb. Mikor azonban több évre visszamenőleg elemezzük az adatokat, amelyekről esetleg nem is tudjuk, ki állította össze és miért, már nagyon hiányozhatnak ezek az információk, mert hiányuk teljesen tévútra vezetheti az elemzőt. A ZR-6 adatrendszerben tárolt adatok elemzésekor azt tapasztaltuk, hogy sok fáradságos detektívmunkát lehetett volna megtakarítani, ha a kísérletezők nem restellték volna megadni azt néhány soros kommentáló szó-

veget, amelyet egyébként kéziratos füzetekben feljegyeztek. Az üzemviteli adatok rendszere ezekből a tapasztalatokból kiindulva helyet biztosít az efféle szöveges információknak, amelyeket az RFIT-ED program nem számszerűen, hanem kártyaképi formátumban tárol.

III. AZ ADATRENDSZER SZERKEZETE

A fenti általános leírás után most rátérünk az adatrendszer szerkezetének a részletes ismertetésére.

III.1. AZ ADATRENDSZER ÁLTALÁNOS SZERKEZETE

Az adatrendszerben tárolt adatok egy mágneslemezen és két velük azonos tartalmu mágnesszalagon tárolódnak. Voltaképpen elegendő lenne az adatrendszer mágneslemezen tárolni, de az ESZR számítógépen a tapasztalat szerint évente többször megsérülnek a mágneslemezekon tárolt adatok, úgy hogy biztonsági okokból van szükség a mágnesszalagokon való duplikált tárolásra. A KFKI számítóközpontjában is minden mágneslemezeről átlagosan havonta készítenek mágnesszalagos másolatokat, ahonnan szükség esetén vissza lehet állítani a lemezen megsérült információt. A mi rendszerünkben az információ duplikálása ettől az általános gyakorlattól eltérően van meg szervezve. Ennek a lényege a következő:

- mágnesszalagra irt másolatok automatikusan készülnek, vagyis a felhasználóknak külön gondjuk ezzel nincs;
- az információ visszaállítását maga az RFIT-ED program végzi.

Ez az utóbbi eltérés első pillanatra jelentéktelen, de - mint a IV. részben látni fogjuk - ez az adatrendszer használatát flexibilissé teszi. Például, a mágnesszalagok felhasználhatók a különböző együttműködő kutatóközpontok közötti adatszere.

Tekintve, hogy a mágnesszalagok a mágneslemezhez képest nem tartalmazzak többletinformációt, a rendszer szerkezetének alábbi ismertetésében figyelmen kívül hagyjuk őket, és az ismertetést annyiban egyszerűsítjük, mintha csak a mágneslemezen tárolódna az információ. A rendszer két file-ból áll: magukat az üzemviteli adatokat tartalmazó file és ennek a katalógusa. Mindkettő közvetlen elérésű szekvenciális file, de az elsőben a rekordok hossza 20 szó (=80 byte), a másodikban 9 szó (=36 byte). Az elsőnek a logikai sorzáma (a KFKI változatnál) 81, a másodiké 80.

Nézzük először a 81-es számú file-nak a belső szerkezetét. Az adatok subfile-okra vannak felosztva. Egy subfile olyan adatokat tartalmaz, amelyek egy adott reaktorblokk meghatározott üzemállapotához tartoznak. Egy adott blokkhoz tartozhat több subfile, amely ugyanolyan üzemállapotra vonatkozik,

de megfordítva már nem lehetséges: egy subfile-ba nem tartozhatnak különböző blokkok vagy üzemállapotok adatai. A subfile-ok azonosítására egy 12 jelből álló jelsorozatot használunk. Ennek a belső felépítése a következő:

P	A	K	S	O	1	O	3	O	1		
1	2	3	4	5	6	7	8	9	10	11	12

ahol

- 1.-4. jel megadja az erőmű nevét (PAKS)
- 5.-6. jel megadja a blokk sorszámát (01)
- 7.-8. jel megadja az üzemállapot sorszámát (03)
- 9.-10. jel szolgál ugyanannak a blokknak azonos típusu subfile-jai közötti megkülönböztetésre (itt: 01, de bármilyen szám lehet 01 és 99 között)
- 11.-12. jel szolgál további megkülönböztetésre, esetleg a subfile titkosítására (általában üresen hagyjuk).

A különböző üzemállapotok megkülönböztetésére az alábbi egyezményes jelölésekben állapodunk meg:

- 01 - fizikai indítás adatai
- 02 - energetikai indítás adatai
- 03 - állandó teljesítmény üzemállapot
- 04 - 99 - később meghatározandó állapotok számára fenntartva.

Megjegyezzük, hogy ezeknek a jelöléseknek akkor lesz jelentőségük, amikor azokat a programokat fogjuk kidolgozni, amelyek a rendszerben tárolt adatokat felhasználják. A jelen pillanatban ezek megadásának még csak tájékoztató jelentősége van. A későbbi felhasználás egyértelműsége szempontjából azonban már most célszerű bizonyos konvenciókat bevezetni, melyik állapotot hogyan jelöljük. Természetesen az RFIT-ED program bármikor lehetővé teszi egy korábbi subfile-név megváltoztatását (lásd IV. rész).

Az atomerőművek jelölésére (1.-4. jel) az alábbi konvenciót vezetjük be: NORD - Nord atomerőmű (NDK), BOHU - Bohunice AE (Csehszlovákia), LOVI - Loviisa AE (Finnország), PAKS - Paksi AE (Magyarország), KOZL - Kozloduj AE (Bulgária), NVOR - Novovoronyezsi AE (Szovjetunió), KOLA - Kolai AE (Szovjetunió), ARMN - Örmény AE (Szovjetunió). Az RFIT-ED program más jelcsoporttal kezdődő subfile-nevet nem fogad el. Természetesen újabb neveknek a felvétele bármikor lehetséges. Annak érdekében, hogy egy új név bevezetéséhez szükséges változtatások keresztülviteléig is lehessen valamilyen fel nem sorolt nevű erőműre is adatokat regisztrálni, a program elfogad egy semleges nevet is: MEAS (az angol measurement - mérés szóból). Például: MEAS020311. Ennek a semleges névnek a használata más esetben is kényelmes lehet, pl. próbaszámítások céljára.

A 11.-12. jelet általában akkor adjuk meg, ha a subfile hozzáférhetőségét korlátozni akarjuk. Ilyenkor mindkét pozícióban az alábbi jelek valamelyikét használjuk:

számok: 0, 1, ..., 9

betük: A, B, ..., Z

egyéb jelek: + - / * . ,

Ha a 11. és 12. jel mindegyike a felsoroltak közül kerül ki, akkor a subfile létrehozásakor a program kiírja, milyen néven lehet ezt a subfile-et a későbbiek során megtalálni. (Ez a név nem szükségképpen azonos azzal, amit a létrehozásakor használtunk). Bármilyen más név megadása esetén a program nem hajlandó a subfile-t létezőnek tekinteni. Tekintve, hogy az adatok tárolása annyira tömör a mágneslemezen, hogy csak az RFIT-ED program segítségével lehet a subfile tartalmát megfejtetni, ez a mód illetéktelen hozzáféréssel szemben elég nagy biztonságot nyújt. Tulajdonképpen két módon lehet az ilyen subfile-okat megtalálni: vagy ismerjük a megtalálásához szükséges nevet, vagy alaposan ismerjük magát az RFIT-ED programot.

A subfile-ok belső szerkezetét a következő pontban (III.2.) ismertetjük. Az e szerkezetnek megfelelően megadott információ 20-szavas rekordok formájában tárolódik a mágneslemezen. Az egyes subfile-ok olyan sorrendben követik egymást, amilyen sorrendben létrehoztuk őket. (Ez a sorrend mágnesszalagokra való másolás, és onnan való visszaállítás útján megváltoztatható, mint a IV. részben látni fogjuk). Annak érdekében, hogy az egyes subfile-ok megtalálásához ne kelljen az egész adatfile-t végigolvasni, a subfile-ok elhelyezkedésére vonatkozó leglényegesebb adatok egy külön file-ban, a katalógusban vannak tárolva. Mint fentebb már mondtuk, ennek a file-nak a logikai száma (a KFKI változatnál): 80.

A katalógusban minden subfile-nak egy 9-szavas rekord felel meg. Mindegyik rekord tartalma:

1.-3. szó: a subfile neve

4.-6. szó: a subfile-t összeállító személy vagy intézmény nevének első 8 betűje (a subfile tartalma alapján, lásd III.2. pont)

7.-8. szó: a legkorábbi és legkésőbbi kampány kezdetének a dátuma (a subfile tartalma alapján, lásd III.2. pont)

9. szó: a subfile-hoz tartozó első rekord sorszáma a 81-es számú file-ban. Ennek a file-nak szerepéről és létrejöttéről a későbbiekben még lesz bőven szó. Itt most annyit jegyzünk meg, hogy a 80-as és 81-es file-ok együttesen alkotják az adatrendszert. Szervezésük emlékeztet a particionált adatrendszerek szerkezetére, jóllehet számítástechnikai szempontból szekvenciális szervezésűek. Az OS operációs rendszer rendelkezik particionált adatrendszerek létrehozásához és kezeléséhez szükséges programokkal. Hogy mégsem ezt használjuk, hanem létrehoztunk egy saját, Fortranban programozott (tehát az operációs rendszer programjaihoz képest lassabban kezelhető) rendszert, annak az a magyarázata, hogy esetünkben a subfile-ok belső szerkezete játssza a fő szerepet, aminek megfelelő programok az OS operációs rendszerben nincsenek.

A rendszer általános szerkezetének az ismertetését a mágnesszalagok tartalmának a leírásával zárjuk. A két mágnesszalag logikai sorszáma (a KFKI változatnál): 60 és 61. Ezek egymással teljesen azonosak. A 80-as és 81-es file-ok tartalmát viszont egyesítve tartalmazzák: a katalógus és a subfile-ok nem külön file-ban, hanem együtt találhatók rajtuk. Kezdődnek az első subfile-hoz tartozó katalógus-rekorddal, ezt követi az első subfile tartalma, ezután jön a második subfile katalógus-rekordja, majd a második subfile tartalma és így tovább a mágneslemezen lévő utolsó subfile-ig.

III.2. A subfile szerkezete

A következőkben ismertetjük a subfile-ok belső szerkezetét, és így az előzőkkel együtt ismertetjük az egész adatrendszer szerkezetét. Az egy subfile-on belül megadható információt csoportokra osztottuk, és mindegyik csoportot megszámoztuk 1-től 99-ig. Amikor az adatokat összeállítjuk, úgy képzeljük el, hogy egy kérdőívet töltünk ki, amelyben számozott kérdésekre válaszolva közöljük az adatrendszer számára megadandó információt. Hogy az egyes kérdésekre adott válaszainkat milyen formátumban kell az RFIT-ED program számára be-lyukasztani, arról a IV. részben lesz szó, itt az egyes kérdések tartalmával fogunk megismerkedni.

A kérdések két nagy csoportra oszlanak: szöveges információ (1-50 kérdések) és számszerű információ (51-99 kérdések). Ez alól egy kivétel van: a 9-es kérdés, amely szintén számszerű adatokat kérdez. Mielőtt a kérdéseket részletesen ismertetnénk, két általános megjegyzést teszünk. Először, nem kötelező minden kérdésre válaszolni, ha egyes kérdésekkel kapcsolatban nincs adatunk vagy közölni valónk. Másodszor, az egyes kérdésekre adott válaszok egymásra épülhetnek, más szóval, bizonyos kérdésekre csak akkor tudunk válaszolni, ha korábbiakra már válaszoltunk. Az RFIT-ED program az ilyen belső összefüggések figyelmen kívül hagyását hibának minősíti, és így meghiusítja formálisan hibás subfile-ok létrehozását.

A) Szöveges információ

Először felsoroljuk azokat a kérdéseket, amelyekre lényegében tetszőleges hosszúságú szöveg megadásával válaszolhatunk (a szöveget célszerű angol nyelven fogalmazni):

1. Annotáció: itt foglaljuk össze, milyen célból állítottuk össze az adott subfile-t. Ha az adott subfile-nak más subfile-okkal valamilyen kapcsolata van, ezt is itt lehet elmondani.
2. Kulcsszavak a subfile tartalmára vonatkozóan: a tudományos-műszaki szakirodalomban használt INIS-descriptorokhoz hasonló kulcsszavak megadására biztosítunk itt lehetőséget. A IV. részben látni fogjuk, hogy az RFIT-ED program biztosítja olyan subfile-ok megkeresését, amelyekben kiválasztott kulcsszavak előfordulnak. Ilyen kulcsszavak használata nagyon megkönnyíti egy sok subfile-t tartalmazó adatrendszerben való eligazodást.

3. Annak a személynek vagy intézménynek a neve, aki vagy ami a subfile-t összeállította. A subfile létrehozásakor ez a név bekerül a subfile katalógusába is (lásd III.1. pont), és a 2. kérdésnél megadott kulcsszavakhoz hasonlóan felhasználható a subfile megkeresésére. Ez az információ lényegében a subfile hovatartozását adja meg arra az esetre, ha felhasználáskor problémák merülnek fel, és tudni szeretnénk kihez forduljunk további információért.
4. Irodalmi hivatkozások az adott reaktorblokkra vonatkozóan.
5. Annak magyarázata, hogy a 9. kérdésnél definiált alcsoportok (lásd ott) minek felelnek meg fizikailag.
6. Irodalmi hivatkozások azokra a mérésekre vonatkozóan, amelyeknek az eredményeit az adott subfile-ban közöljük.
- 7.-8. Egyelőre nincsenek definiálva.
9. A subfile-ban később közölt adatok alcsoportjainak a meghatározása. Mint később látni fogjuk, a számszerűen közölt adatok (51.-99. kérdések) reaktorkampányok szerint alcsoportokra vannak osztva. Egy adott reaktorblokkra vonatkozóan mért adatok természetszerűen bomlanak fel a különböző kampányokhoz tartozó adatokra. Egy kampányon belül az adatok egymással szoros kapcsolatban vannak, de a különböző kampányokat egymás között csak az átrakási séma és az egyes kazettákban korábban felhalmozódott kiégés mértéke köti össze, tehát a kapcsolat közöttük elég laza. Így a számszerű információt a kampányok természetes módon alcsoportokra osztják, és ezt a felosztást megtartjuk a subfile-on belül is. A subfile-on belül az információt tartalmi szempontból a kérdések már egy módon csoportokra osztják, de minden csoporton belül a kampányok szerint egy további felosztást kapunk. Ezt neveztük az előbb alcsoportoknak. A legegyszerűbb úgy eljárni, hogy az első alcsoport a blokk első kampányának felel meg, a második alcsoport a második kampánynak és így tovább. Az RFIT-ED program eleve ezt a szerkezetet tételezi is fel. Nincs ok azonban arra, hogy ezt kötelezően elő is írjuk: előfordulhat, hogy az alcsoportok sorrendje nem követi a kampányok sorrendjét, vagy, hogy egy kampányhoz tulságosan sok adat tartozna, és emiatt jobb egy kampányt két alcsoportban megadni stb. (Egy alcsoport-hoz megadható adatok száma bizonyos kérdéseknél korlátozva van.) Ezért lehetőség van arra, hogy a 9. kérdéshez megadott válasunkkal az alcsoportokat tetszőlegesen definiáljuk. Egy alcsoportot úgy definiálunk, hogy megadjuk a kampány kezdődátumát és a kampány nevét. Ahány alcsoportot itt definiálunk, annyi alcsoport-ra vonatkozóan vár az RFIT-ED program a későbbi kérdéseknél adatot. A dátum megadása a következőképpen történik: pl. 1981. október 13-ának felel meg a 131081 szám megadása, tehát nap-hónap-év egy hatjegyű egész szám formájában. (A hónap sorszáma mindig kétjegyű: pl. májusnak 05 felel meg!) Az itt megadott dátum lesz a subfile-ban a kampányra megadott minden időpont kezdőpontja. A program elfogadja a dátum elhagyását: ilyenkor helyette nullát vesz. A dátum elhagyása megnehezíti a felhasználó dolgát, tehát elég udvariatlan dolog.

Az itt megadott dátumok alapján tölti fel a program a katalógus 7. és 8. szavát is. A kampány neve egy tetszőleges 8 jelből álló jelcsoport lesz. Megadása szintén elhagyható: ilyenkor a program ad nevet az alcsoportnak. Az első alcsoport neve CAMPGNO1 lesz, a másodiké CAMPGNO2 és így tovább. Az elmondottakból nyilvánvaló, hogy a 9-es kérdéshez tartozó információ alapvető a számszerű adatok szerkezete szempontjából. Ezért az RFIT-ED program az 51-es kérdéstől kezdve csak akkor fogad el adatokat, ha a 9-es kérdésre kapott értelmezhető választ.

10. Megjegyzések a hőteljesítmény meghatározásával kapcsolatban: annak megadásáról van szó, hogy milyen módszerrel határoztuk meg (ionizációs kamra, szekundérköri hőmérleg stb.).
11. Megjegyzés a bórsavkoncentráció mérésével kapcsolatban.
12. A reaktorblokk hőtechnikai sajátosságai: a II.2. pontban volt szó róla, hogy a rendszer csak neutronfizikai adatokat tárol. Az ezek értelmezéséhez szükséges termohidraulikai adatokat itt lehet szövegszerűen megadni (a hűtőközeg hozama, a kazettafalak között áramló hányad és hasonlóak). Fel kell azonban hívni a figyelmet arra, hogy az itt megadott számszerű adatokat - tekintve hogy szöveggként tárolódnak - más programok nem használhatják számszerűen.
13. Megjegyzések azokkal a teljesítményváltozásokkal kapcsolatban, amelyek esetleg nem látszanak az 51-es kérdésnél megadott táblázatban.
14. Megjegyzések az 51-es kérdésnél megadott mátrixok oszlopainak a megadásával kapcsolatban. Különösen azt célszerű itt megmagyarázni, miért nem adjuk meg némelyiket (ha van ilyen).
15. Megjegyzések a megadott mérési hibákkal kapcsolatban.
16. Megjegyzések a reaktivitáseggyűtthetők mérésével kapcsolatban.
17. Megjegyzések a rudértékeség mérésével kapcsolatban.
18. Megjegyzések az átrakásokkal kapcsolatban: milyen módszerrel optimalizáltuk, esetleg inhermetikusságok előfordulása stb.
19. Megjegyzések a teljesítménykiválás eloszlásainak a meghatározásával kapcsolatban.
20. A hőmérsékleteloszlások meghatározására vonatkozó megjegyzések.
21. Megjegyzések az in-core detektorokkal kapott mérési eredményekre vonatkozóan.

* * * * *

A 22.-50. kérdések tartalma még nincs definiálva. Ha felmerül a későbbiekben igény egy további kérdés bevezetésére, ezek a sorszámok felhasználhatók erre.

Mint fentebb mondtuk, bármelyik kérdésre adott válasz elhagyható. Ezért olyan közlések, hogy "nincs megjegyzés" feleslegesek, mert ezt pontosan kifejezi a kérdés válasz nélkül hagyása. Egy meglévő subfile-ba bármikor lehet pótlólag információt beépíteni. Az üzemviteli adatok rendszere a kiindu-

lástól kezdve nemzetközi együttműködésben jön létre. Ez az oka annak, hogy az angol nyelv használatát javasoljuk a szöveges kérdések megválaszolásánál. Az RFIT-ED program outputja is angol nyelvű. Ez nem jelenti persze azt, hogy nem használhatunk bármilyen más nyelvet, amely a rendelkezésre álló jelkészlettel (ami éppen az angol ábécé) leírható.

B) Számszerű információ

A számszerű információ az 51-es és további kérdésekre adott válaszokban írható be az adattároló rendszerbe. Minden kérdésen belül az adatok a 9-es kérdésnél meghatározott alcsoportra oszlanak, így a számszerű információ megadásának előfeltétele a 9-es kérdés megválaszolása. A 9-es kérdésnél elmagyaráztuk, mi a különbség a "kampány" és az "alcsoport" között. Az ott elmondottak szem előtt tartásával az alábbiakban az alcsoport helyett a kampány kifejezést használjuk, mert az alcsoport kifejezés erőltetése feleslegesen mesterkéltté tenné a szöveget.

51. Ennél a kérdésnél adjuk meg azokat a mennyiségeket, amelyeket a II.2. pontban alapmennyiségeknek neveztünk. Minden a 9-es kérdésnél definiált kampányra megadunk egy mátrixot, amelynek minden sora egy kritikus állapotnak felel meg. A mátrix egyes oszlopainak a fizikai jelentése:

- 1 - a kritikus állapot sorszáma
- 2 - a naptári nap attól az időponttól számítva, amelyet a 9-es kérdésnél dátumként megjelöltünk
- 3 - az adott állapotig eltelt effektív napok száma (napban)
- 4 - a reaktor hőteltjesítménye (%-ban)
- 5 - a szabályozó szerv (általában a K_6 csoport) helyzete (cm-ben)
- 6 - borkoncentráció (g/kg-ban)
- 7 - a hűtőközeg bemenő hőmérséklete ($^{\circ}\text{C}$ -ban)
- 8 - átlagos felmelegedés az aktív zónában ($^{\circ}\text{C}$ -ban)
- 9 - a primérgör nyomása (MPa-ban)
- 10 - annak a csoportnak a száma, amelyre az 5. oszlop vonatkozik
- 11 - a hűtőközeg kilépő hőmérséklete ($^{\circ}\text{C}$ -ban): a 7. és 8. oszlop összege

A mátrix tetszőleges oszlopát el lehet hagyni attól függően, hogy melyiket tartjuk célszerűnek megadni a konkrét esetben. A megadott oszlopok száma nem haladhatja meg a 10.-et. Ennek a mátrixnak a megadása elengedhetetlen ahhoz, hogy a későbbi információt értelmezni lehessen. Az itt megadott kritikus állapotokra a subfile többi részében az állapot sorszámaival hivatkozunk. Ez a sorszám rendszerint a mátrix megfelelő sorának a sorszáma. Ettől azonban el is térhetünk, ha az első oszlopban sorszámokat adunk meg. Ezek már tetszőleges egész számok lehetnek. Ilyen, a természetestől eltérő sorszámozásra ritkán van szükség, de előfordul. Vegyük például azt az esetet, amikor összeállítottuk a

subfile-t a mátrix első oszlopa nélkül, és az 51-es utáni kérdéseknél a természetes módon hivatkoztunk a kritikus állapotokra. Utóbb kiderül, hogy a mátrix megadása hibás volt, és bizonyos sorokat törölni kell, másokat meg a meglévők közé be kell iktatni. Ekkor a javított mátrixban bizonyos állapotok sorszáma megváltozhat. Ha később sok hivatkozás van már a subfile-ban az eredeti sorszámokra, célszerű nem a hivatkozásokat átjavítani, hanem a mátrix első oszlopát pótlólag feltölteni, hogy a már meglévő hivatkozások jók maradjanak a javítások után is.

52. Az 51-es kérdésnél az egyes kampányokhoz megadott mátrixok adatainak a mérési hibái, ha ezek egy paraméterrel jellemezhetők: a hiba minden sorban ugyanaz vagy relatív hiba minden sorban ugyanaz (l. még 54-es kérdést is). A hibák megadásának az eseteit a II.1. táblázatban foglaltuk össze. Az 52-es kérdésnél a hiba megadási módját jelző változó és hiba számításához szükséges paraméter értékét kell oszloponként megadni. Megjegyezzük, hogy itt átvettük a ZR-6 adatrendszernél használt eseteket.
53. Az 51-es mátrixokban megadott mennyiségek egységei, ha azok nem a fent megjelölt egységekben vannak megadva. Itt oszloponként egy-egy olyan szorzótényezőt kell megadni, amellyel az 51-es mátrixokban megadott értékeket meg kell szorozni ahhoz, hogy azokat a szokásos egységekben kapjuk. Ha valamelyik oszlopra nem adunk meg ilyen tényezőt, vagy valamilyen kampányra egyáltalán nem adunk meg ilyen adatokat, akkor ezzel azt jelezzük, hogy a megfelelő adatok a szokásos egységekben vannak megadva.
54. Az 51-es mátrixokban megadott mennyiségek mérési hibái, ha azok soronként változnak. Ez azt jelenti, hogy megvan az a (ritkán szükséges) lehetőség is, hogy a mátrix egyes oszlopaiban megadott minden számhoz külön-külön rendeljünk hibát. Ilyesmire akkor lehet például szükség, ha valamelyik mennyiség mérésének a módszere a kampány során megváltozik. Ilyen esetben azonban nagy gondosságra van szükség: amikor megváltoztatjuk az 51-es mátrixot (például új sorokat iktatunk be, vagy néhányat kitörlünk), ezzel összhangban kell az itteni információt is változtatni kell, különben a hibák és mért értékek összetartozása hibás lesz. Az RFIT-ED programnak nincs módja az ilyen természetű változtatások helyességét ellenőrizni.
55. Az egyes kampányok kartogrammái. Rendszerint minden kampányhoz egy kartogrammat adunk meg (bár a program elfogad kampányonként többet is), de nincs előírva, hogy minden kampányhoz adjunk meg kartogrammat (nem is lenne értelme: elképzelhető, hogy a kartogrammat egy másik subfile-ban az adott kampányhoz már megadtuk). A kartogramma megadásának a módja megfelel annak, ahogy azt a BIPR programban szokás megadni. A kazetták szabványos számozásának megfelelően sorban megadjuk az alábbi számokat:
 - az első kampány esetében minden pozícióra megadjuk a dúsítások konvencionális jelét: az 1,6%-ra 1-et, a 2,4%-ra 2-t, 3,6%-ra 3-at;

- a további kampányoknál a kazetta által az előző kampányban elfoglalt pozíció sorszámát (ha átrakott kazetta), a dusítás konvencionális jelét (ha a kazetta friss), tehát rendre 1001, 1002 és 1003-at.

Könnyítésül a kartogramma megadásánál figyelembe vehetünk bizonyos szimmetriákat. A megadott pozíciók száma elégéséges a szimmetria meghatározására. Ezek az alábbi táblázatból olvashatók ki:

pozíciók száma	szimmetria
349	360°
184	180°
117	120°
68	60°
37	30°

Későbbi kérdéseknél (69, 70, 71) a kazettánként kiégés értékeit fogjuk megadni. Ezeket az eloszlásokat az itt megadott kartogrammához rendeljük, tehát az ott megadandó adatok száma meg kell hogy egyezzen az itt megadott számmal. Ebben az értelemben az 55-ös kérdés feltétele a későbbieknek. A zóna szimmetriáját ugyanebből az okból sem célszerű egy subfile-on belül megváltoztatni. Ha a szimmetria üzemviteli okokból megváltozott, akkor az áttekinthetőség kedvéért jobb az adott reaktorblokkra egy új subfile-t nyitni.

56. A termopárok helyzete: ez egy adott blokk esetében rögzített, ezért ezt elegendő egy kampányra vonatkozóan megadni (célszerű mindjárt az első kampányra vonatkozóan). Ez aztán minden további kampányra érvényes marad. Fontos felhívni a figyelmet arra, hogy ennek összhangban kell lennie a kartogramma megadott szimmetriájával (55-ös kérdés), tehát ha a kartogramma szimmetriája megváltozott, a termopárok helyzetét újból meg kell adni az új szimmetriának megfelelően. Mint az 55-ös kérdésnél megjegyeztük, ez általában egy új subfile megnyitásával jár együtt. A termopárok helyzetét úgy adjuk meg, hogy felsoroljuk azoknak a pozícióknak a sorszámát, amelyekben termopárok vannak.
57. Az in-core detektorok helyzete: megadásuk ugyanugy történik, mint a termopároké. Az 56-os kérdésnél tett megjegyzések ide is vonatkoznak.
58. A szabályozó szervek helyzete: megadásuk ugyanugy történik, mint a termopároké. Az 56-os kérdésnél tett megjegyzések ide is vonatkoznak.
59. A kazetták szállítási adatai: ezek természetesen csak a 360°-os szimmetriára vonatkozhatnak. Ezért ez a kérdés formálisan független az 55-ös kérdésnél megadott kartogrammáktól. (Tartalmilag természetesen összefüggnek.) Minden friss kazettára a következő adatokat kell megadni:

1. a kazetta gyártási száma
2. az urán tömege (grammban)
3. az ^{235}U tömege (grammban)
4. a pozíció száma a 360° -os kartogrammában
5. a kazetta orientációja (0° , 60° , 120° , ...)

Ha az adatok másképp vannak megadva (többet vagy kevesebbet adtunk meg, más sorrendben adtuk meg stb.), akkor erről a 18-as kérdésnél magyarázatot kell adni. Az ennél a kérdésnél megadott információ felhasználható például arra, hogy a subfile-ban megadott információ alapján adatot szolgáltatassunk a NAÜ számára.

60. A reaktivitástényezők méréseinek az eredményei: egy az 51-esen megadott mátrixokhoz hasonló mátrix alakjában adjuk meg az adatokat, de az egyes oszlopok jelentése más:

1. az állapot sorszáma (az 51-es kérdésnél megadott mátrix számozása szerint), amelyhez az adott sor tartozik;
2. $\partial\rho/\partial C_B$
3. a 2-es oszlopban közölt érték mérési hibája
4. $\partial\rho/\partial T$
5. a 4-es oszlopban közölt érték mérési hibája
6. $\partial\rho/\partial N$ (teljesítménytényező)
7. a teljesítménytényező mérési hibája
8. $\partial\rho/\partial H$
9. a 8-as oszlopban közölt érték mérési hibája
10. annak a szabályozó csoportnak a száma, amelybe a 8-as oszlopban közölt érték vonatkozik.

Mint az 51-es kérdésnél, itt is tetszőleges oszlop elhagyható.

61. A 60-as kérdésnél megadott adatok egységei, ha azok nem a szokásosak. A 60-as és 61-es kérdések közötti kapcsolat analóg az 51-es és 53-as kérdések közöttivel.

62. A szabályzó rudak értékségei. Az 51-es kérdéshez hasonló módon a következő oszlopokból álló mátrixokat kell megadni:

1. Az állapot sorszáma (az 51-es kérdésnél megadott mátrix számozása szerint), amelyhez az adott sor tartozik;
2. annak a reaktivitásnak az értéke, amelyet akkor kapnánk, ha a tekintett rudat kihuzzuk;
3. a 2-es oszlopban közölt érték mérési hibája;
4. a tekintett rudcsoport száma;
- 5.-10. a csoport egyes rudjainak a helyzete, ha a 2. oszlopban megadott reaktivitás nem az egész csoportra vonatkozik.

Ha az adott kritikus állapotból kiindulva egymás után több rudat huzunk ki, akkor a 2-es oszlopban az adott rudcsoportra és az előzőleg már kihuzott rudcsoportra vonatkozó reaktivitásértékségek összegét kell megadni, a kritikus állapot sorszáma végig a kiindulási állapot sorszáma.

Ha az egyes rudcsoportok értékségét minden sorban külön adjuk meg (tehát nem az előzőkkel összeadva), akkor ezt csak úgy tehetjük meg, hogy minden előzőleg tekintett rud kihuzott helyzetét az 51-es mátrixban mint külön kritikus állapotot definiáljuk.

63. A 62-es kérdésnél megadott adatok egységei, ha azok nem a szokásosak. Magyarázatot lásd az 53-as kérdésnél.
- 64.-68. Egyelőre nincsenek definiálva.
69. A kazettánkénti kiégés eloszlása a hűtőközeg felmelegedésének a mért értékei alapján. Ennél a kérdésnél az 51-es mátrixokban definiált néhány állapotra megadjuk minden kazettára az addig az állapotig a kazettában a kiégés értékét MWnap/tonna egységekben. A megadandó értékek számát a program az 55-ös kérdésből veszi, tehát a kartogramma minden pozíciójára valamelyik értéknek szerepelnie kell. Ha az eloszlás értéke valamelyik pozícióban nem áll rendelkezésre, nincs más kiút mint megállapodásszerűen 0-t megadni. Megjegyezzük, hogy ilyen eloszlásokat a program nem fogad el, ha az adott kampányra az 55-ös kérdésnél nem adtunk meg kartogrammat. Ha egy kazetta az adott kampányban már nem friss, akkor itt az előző kampányokból származó kiégést is figyelembe kell venni. A 19-es kérdésnél utalni kell arra, ha ettől a megállapodástól eltérünk.
70. A kazettánkénti kiégés eloszlása nukleáris mérés alapján. Lásd a megjegyzéseket a 69-es kérdésnél.
71. A kazettánkénti kiégés eloszlása számítás alapján. Lásd a megjegyzéseket a 69-es kérdésnél.
72. Hőmérsékleteloszlások: megadásuk analóg a 69-es kérdésnél mondottakkal, de a pozíciók számát az 56-os kérdésekből vesszük.
73. Az in-core detektorokkal végzett mérések eredményei: a megadásuk analóg a 69-es kérdésnél mondottakkal, de a pozíciók számát az 57-es kérdésből vesszük. Ezeknél az adatoknál az is eltérés, hogy itt egy pozícióban egy-nél több értéket kell megadni, mert pozícióként több in-core detektor helyezkedik el.
- 74.-99. Egyelőre nem definiált kérdések.

* * * * *

A subfile-ok tartalmának az ismertetését néhány megjegyzéssel zárjuk.

- A fentiekből következik, hogy az alábbi kérdések között vannak tartalmi összefüggések:

51 - 52

55 - 56, 57, 69, 70, 71

56 - 72

57 - 73

Ez gyakorlatilag azt jelenti, hogy az elől álló kérdések megválaszolása feltétele a hátul állók megválaszolásának.

- Könnyen elképzelhető, hogy a rendszer a jelenlegi formájában nem a legoptimálisabb felépítésű. Amint egyre több atomerőműből fognak rendelkezésre állni adatok, az ezekkel szerzett tapasztalatok vezethetnek olyan következtetésre, hogy a rendszert némileg módosítani kell. Az RFIT-ED program jelenlegi állapotában erre minden további nélkül van lehetőség némi programozói munka árán. Olyan dolgokra gondolunk, mint például a termoelemek elhelyezkedésének a szimmetriája. A jelenlegi rendszer feltételezi, hogy ezek olyan szimmetria szerint helyezkednek el, mint a kartogramma szimmetriája. Ha az egyes blokkokra nem igaz, akkor az 55-ös és 56-os kérdéseket függetleníteni kell egymástól.
- A Nord Atomerőmű néhány adata, továbbá egyes szovjet adatok nem férnek be a fentebb az 51-es kérdésnél ismertetett mátrixba: ezek miatt növelni kell majd az oszlopok számát, és el kell ejteni azt a jelenleg érvényes megszorítást, hogy az oszlopok száma legfeljebb 10 lehet. A kísérleti adatok sokszínűsége valószínűleg a jövőben is szükségessé fog tenni hasonló kisebb változtatásokat, tehát időnként - mikor több ilyen igény felmerült - egy programrevíziót kell végrehajtani. Nagyon fontos azonban, hogy ezek ne érintsék az előzőleg a rendszerbe felvett adatokat.

II.1. Táblázat. Hibák megadásának a módja (l. 52-es és 54-es kérdés)

vezérlő változó	jelentés	paraméter	megjegyzés
0	hiba egyenlő 0	-	-
1	Poisson hiba	-	ritka
2	állandó hiba	σ^2	-
3	változó hiba	-	hibák 54-ben
4	állandó relatív hiba	$(\sigma/y)^2$	-

IV. AZ RFIT-ED PROGRAM HASZNÁLATA

IV.1. ÁLTALÁNOS MEGJEGYZÉSEK

A III.1. pontban ismertettük, hogy az adattároló rendszer a 80-as és 81-es logikai számú diszk-file-okban és a 60-as és 61-es logikai sorszámú mágnesszalag-file-okban található. Amikor a rendszerrel valamilyen műveletet hajtunk végre, még további file-okat is használunk. Ezek ideiglenes diszk-file-ok, amelyek az RFIT-ED program futásának a befejezése után törlődnek. Ezeknek az ideiglenes file-oknak a logikai sorszáma: 74, 75 és 77. Használatukat több szempont is indokolja:

- Amikor egy új subfile-t hozunk létre, igen gyakoriak az input-hibák. Ezek eredményeképpen hibás vagy nem teljes subfile-ok keletkeznek. Ha a program az új subfile-t rögtön az állandó file-okon (80 és 81) készítené, akkor ezeken könnyen keletkezhetnek használhatatlan "hulladék-subfile-ok", amelyekből a rendszert folyton tisztogatni kellene. Erre természetesen az RFIT-ED program nincs felkészülve, tehát ez külön gondot okozna. Jobb ezért az új subfile-okat először ideiglenes file-okban szerkeszteni, ahonnan sikeres befejezés esetén az új subfile-t át lehet másolni az állandó file-okba. Hasonló megfontolások érvényesek meglévő subfile-ok javítására is.
- Az a tény, hogy subfile-ok átvihetők ideiglenes file-okba is, lehetővé teszi, hogy egyes subfile-okon úgy is lehessen változtatni, hogy az állandó file-okat érintetlenül hagyjuk. Ennek akkor van jelentősége, amikor használjuk a subfile-t, de nem eredeti alakjában, hanem bizonyos változtatásokkal, viszont nem vagyunk annyira biztosak e változtatások helyességében, hogy ezeket véglegesen is végrehajtsuk. A ZR-6 adatrendszerrel szerzett tapasztalatok egyértelműen igazolják ennek a lehetőségnek a hasznosságát. Megjegyezzük, hogy az OS operációs rendszer a programok javítását is megengedi ideiglenes file-okon vagy véglegesen.
- Vannak bizonyos javítási műveletek, amelyek csak ideiglenes file-ok segítségével lehetségesek. Ennek részletesebb magyarázatához olyan programozástechnikai részletekbe kellene bocsátkoznunk, amelyek kívül esnének a jelen dolgozat keretein, ezért ettől eltekintünk.

Mindezt előrebocsátva megérthetjük azokat a műveleteket, amelyeket az RFIT-ED program lehetővé tesz. A jelenlegi változatban a következő műveletek lehetségesek:

- új subfile-ok létrehozása: ezek mindig a 77-es file-on keletkeznek;
- meglévő subfile-ok javítása: a javított változat mindig a 77-es file-on keletkezik, de az utoljára javított subfile javított változata létezik még (a következő subfile-javításig) a 75-ös file-ban is; javíthatunk az ideiglenes file-on (77) és az állandó file-okon található subfile-okat;
- subfile-ok felvitele a 77-es file-ról az állandó file-ba;
- subfile-ok tartalmának a listázása; listázhatók subfile-ok teljes egészében vagy részlegesen; listázhatók az ideiglenes és az állandó file-okban található subfile-ok;
- subfile-ok törlése az ideiglenes vagy az állandó file-okból;
- subfile-ok másolása a diszk-file-okról (80 és 81) a mágnesszalagokra és megfordítva; itt a lehetséges műveleteknek egy elég gazdag családjáról van szó, amelyek az egész adatrendszert kényelmesen kezelhetővé teszik;
- a meglévő subfile-ok neveinek a listázása;
- subfile-ok átnevezése;
- van végül néhány speciális művelet, amelyek bizonyos kényelmi okokból váltak szükségessé.

Mindezeket a műveleteket a program általában lyukkártyán megadott input adatok alapján hajtja végre. Minden művelethez az input adatokban egy utasítás tartozik. Az utasítások 4 betűből álló kulcsszavak, amelyek a művelet angol nevének könnyen megjegyezhető rövidítései. A program az input adatokat utasítástól utasításig olvassa. Egy utasítás végrehajtására csak akkor kerül sor, amikor az előző utasítás minden adatát a program feldolgozta, és azt az utasítást végrehajtotta. Ily módon az egyes utasításokhoz tartozó adatok egymástól világosan elválasztódnak, ami az adatok összeállítását jelentősen megkönnyíti. Ugyanez teszi lehetővé, hogy az egyes utasításokhoz tartozó adatok megadásának a szabályait egymástól függetlenül írjuk le. A használható utasításokat a későbbiekben ismertetjük.

IV.2. AZ INPUT ADATOK ÁLTALÁNOS SZERKEZETE

Az input adatok általában három részből állnak:

- (1) az utasítást tartalmazó kártya,
- (2) ezt követi a subfile kijelölése, tehát annak megadása, hogy az utasítás melyik subfile-ra vagy subfile-okra vonatkozik,
- (3) végül a többi adatok, amelyek összetétele függ magától az utasítástól.

Vannak utasítások, amelyeknél akár a subfile-kijelölés, akár a többi adat, vagy akár mindkettő elmaradhat. Ennek az általános szerkezetnek a megvilágítására mutatjuk az alábbi példát (a benne használt NEW, CORR, DEL7, LIST, RCRD, END utasítások értelmét a későbbiekben találjuk meg):

NEW

PAKS010101

.....

adatok ehhez a subfile-hoz

.....

PAKS0102101

.....

adatok ehhez a subfile-hoz

.....

két új subfile létrehozása
a 77-es ideiglenes file-on

CORR

PAKS010201

.....

adatok ennek a subfile-nak
a javításához

.....

a második létrehozott subfile
javítása

DEL7	}	a javított subfile első változatának a törlése
PAKS010201		
LIST		
PAKS010101	}	a két subfile listázása
PAKS010201		
RCRD	}	a két új subfile másolása az állandó file-ba
END		

Az utasításokat, a subfile-ok nevét és általában a kulcsszavakat (azokat is, amelyeket a továbbiakban fogunk megismerni) mindig a kártya bal szélén kezdve kell lyukasztani, mert különben a program nem ismeri fel őket. Ettől eltekintve az adatok lyukasztása elég kötetlen, gyakorlatilag formátum nélküli. Mielőtt az egyes utasítások pontos értelmét, és a hozzájuk tartozó adatok megadásának a szabályait ismertetnénk, szükséges még néhány általános szabályt megismerni, amelyek minden utasításra egyformán érvényesek.

A) Különböző adathordozók közötti átkapcsolás (PERI)

Az eddigiekben úgy fejeztük ki magunkat, hogy az "input adatokat lyuk-kártyán adjuk meg". A továbbiakban is mindig inputkártyákról fogunk beszélni, de ez nem jelenti azt, hogy az adatoknak ténylegesen kártyán is kell megadva lenniük. Igen gyakran előfordul ugyanis, hogy az adatok nem kártyán, hanem lyukszalagon vagy diszken állnak rendelkezésre, sőt az sem ritka, hogy az adatok különböző adathordozókon vannak meg. Nos, az RFIT-ED program megengedi hogy az input adatok olvasását bármelyik helyen egy bizonyos periférián megszakítsuk, és egy másikon folytassuk. Ennek az a módja, hogy a kívánt helyen megadunk egy

PERI k

tartalmu "kártyát", és ennek hatására az olvasás "k" logikai sorszámú periférián folytatódik. Az adatoknak természetesen az új adathordozón is kártyakép formátumban kell megadva lenniük. A kártyaolvasó logikai szám: 5 (OS szabvány). A program mindig feltételezi, hogy az adatok ezen a periférián kezdődnek, de ezután megszorítás nélkül össze-vissza ugrálhatnak a különböző adathordozókon. Közben persze akárhányszor visszakapcsolhatunk a kártyaolvasóra is egy-egy

PERI 5

sor megadásával.

B) Az input adatok nyomtatása (INPT)

Igen gyakran célszerű nyomonkövetni, milyen input adatokat olvasott a program. Ez különösen akkor van így, amikor többször váltottunk perifériát, vagy amikor hibát keresünk az adatainkban. Ezért az RFIT-ED programban lehetőség van az input kártyák tartalmának a kinyomtatására. Ha ezt igényeljük,

akkor az utasítást tartalmazó kártya után közvetlenül (de még a subfile azonosítás előtt) adjuk meg az

INPT

kulcsszót, aminek eredményeképpen a következő kulcsszóig minden adatkártya tartalmát a program ki fogja nyomtatni, mielőtt a kártya feldolgozásához hozzáférne. Minden utasítás megszünteti az előző utasításhoz megadott INPT kulcsszó hatását.

C) Számok megadásának a formátuma

Az RFIT és RFIT-ED programokhoz egy saját input-rendszert fejlesztettünk ki, mert a Fortran IV inputrendszere elégtelennek bizonyult. Ez utóbbinak a fő hiányossága az, hogy az adatok megadása tulságosan kötött, márpedig az adattároló rendszer olyan, hogy az inputadatok szerkezete gyakran függ a korábbi adatoktól (még egy kártyán belül is). Ezen túlmenően nem lehet senkitől elvárni, hogy az itt szükséges mennyiségű számot kötött formátumban hiba nélkül meg tudjon adni. Ezért e programok számára a számokat teljesen kötetlen formátumban lehet megadni. Például a 12.34 valós számot a következő módok bármelyikén meg lehet adni:

12.34	1.234E+1	1.234+1	0.1234E+2
1.234D+1	0.1234+2	123.4-1	stb.

A különböző számok között tetszőleges számú betűközt hagyhatunk (de legalább egy szükséges). Egy számnak a végét a következők jelölhetik: betűköz, kártya vége, betű (kivéve E és D), vessző vagy bármilyen más nem numerikus jel. (Numerikus jelek: számjegyek, tizedespont, plusz, mínusz, E és D.) A kitevő részben E és D után tehetünk betűközt. A program tulajdonképpen minden logikusan megfejtendő számot megfejt. Végül megjegyezzük, hogy minden számot először valós számként olvas el, és ezt egész számmá váltja át, ha az adott helyen a szám értelemszerűen egész szám. (Átváltáskor a szokásos módon kerekít.)

Az adatoknak ez a formátumnélkülisége nemcsak a program felhasználói számára jelent könnyebbséget, hanem a program megírását is megkönnyítette, és olyan inputszabályokat tett lehetővé, amelyekről a szokványos Fortranban szó sem lehetne. Ennek azonban ára van: ebben az inputrendszerben a program tetszőleges szövegen belül megtalálja a számokat, és így hibásan megadott adatok esetében olyan helyen is olvashat számokat, ahol az adatok összeállítója nem is várná. Emiatt különös fontossága van annak, hogy a program minden lehetséges módon ellenőrzi, hihetők-e a megadott adatok.

D) Adathibák

Amikor nagy fáradsággal összeállítjuk input adatainkat, amelyekből a programnak subfile-okat kell szerkesztenie, sokféle hibát véthetünk. E hibák felderítésére az embernek és a számítógépnek különböző esélyei vannak. Amikor egy adatmátrixra ránézünk (pl. arra, amelyet az 51-es kérdés kapcsán adunk

meg), egy szépen kinyomtatott listán könnyen észrevevesszük, hogy valamelyik szám értéke hibás: nem nehéz észrevenni, például, hogy a reaktor teljesítménye nem lehet 950%. Könnyen rájövünk, hogy 95%-ot akartak megadni, de a tizedes pontot rossz helyre lyukasztották. Efféle hibák felderítésére a gépnek az emberhez képest összehasonlíthatatlanul kevesebb esélye van (bár elvileg nem lehetetlen). Más a helyzet azonban akkor, ha arról van szó, hogy több száz adatkártyát tekintve megvan-e rajtuk a mátrix minden eleme, vagy esetleg több van-e belyukasztva. Az efféle hibák felderítésében a gép előnyben van az emberrel szemben. Ilyen megfontolások alapján az RFIT-ED program megírásakor arra törekedtünk, hogy maga a program minél több formális és strukturális adathibát vegyen észre, de nem törekedtünk arra, hogy a megadott fizikai mennyiségek értékeit hihetőség szempontjából megvizsgálja. Ez utóbbi vizsgálat számára készülnek a subfile-okról a listák, amelyekben az emberi szem jobban eligazodik.

Formálisan hibás adatok a szokványos Fortran-inputban előbb-utóbb futási hibát eredményeznek, amelynek az eredményeképpen a program futása megáll, és a hibás hely után következő adatok olvasására már nem kerül sor. Az RFIT-ED sajátos inputrendszere kiküszöböli ezt a nehézséget: adathiba miatt a program futása nagyon ritkán szakad meg. A hibás helyeket a program úgy deríti fel, hogy az utána következő esetleg hibátlan adatok feldolgozására még sor kerül. Azok, akik a RFIT vagy RFIT-ED programokat használják, tapasztalják, hogy ezek a programok gyakran irnak ki hibaüzeneteket, amelyek alapján viszonylag könnyen meg lehet találni a hiba forrását. Maguk az üzenetek többnyire önmagukban érthetők, de azok számára, akik további magyarázatot igényelnek, a [4] riportot ajánljuk. Bennük bizonyos tanácsokat is találhatunk, hogyan reagáljunk a különböző hibaüzenetekre.

IV.3. SUBFILE-OK KIJELELÉSE

A IV.2. pontban volt szó arról, hogy az input adatok egyik része azoknak a subfile-oknak a kijelölése, amelyekre a különböző utasítások vonatkoznak. Az alábbiakban ennek a részleteit ismertetjük. Az RFIT-ED program két fajta subfile-kijelölést enged meg: explicit és implicit.

Az explicit subfile-kijelölés azt jelenti, hogy megadjuk a kijelölendő subfile-ok nevét (lásd III.1). A subfile-ok neve (pontosabban bennük az atomerőmű jele) kulcsszónak számít, tehát a program csak akkor fogja subfile-névként felismerni, ha a kártya bal szélére lyukasztjuk. Ebből következik, hogy subfile-neveket felsorolásban is mindig külön kártyára kell lyukasztani.

Az implicit subfile-kijelölés arra szolgál, hogy a kijelölendő subfile-ok nevének az ismerete nélkül, a subfile-ok tartalma alapján jelölhessük ki őket. Ez gyakran csak kényelmi lehetőség, néha azonban szükségszerű: éppen arra vagyunk kíváncsiak, melyek azok a subfile-ok, amelyek bizonyos feltételeknek eleget tesznek. Az implicit subfile-kijelölés formája a következő:

ITEM A k_1, k_2, k_3, \dots

ahol ITEM kulcsszó, tehát a kártya baloldalára kell lyukasztani, A valamilyen

ismertetőjelnek a betűjele, k_1, k_2, k_3, \dots paraméterek, amelyek értéke és száma A konkrét értékétől függ. Az A paraméternek az alábbi értékei lehetnek:

B - kijelölés a reaktorblokk szerint

k_1 : az atomerőmű betűjele

k_2 : a blokk sorszáma az atomerőműn belül (el is hagyható: ekkor az erőmű minden blokkját kijelöljük)

k_1 és k_2 értékeit a subfile-ok nevéből veszi a program.

T - kijelölés a subfile típusa szerint (a subfile nevének 7. és 8. jegye)

k_1 : a kívánt típus sorszáma

D - kijelölés dátum szerint

k_1 : kezdő dátum

k_2 : végső dátum

Kijelölődnek azok a subfile-ok, amelyeken a 9-es kérdésben k_1 és k_2 közé eső dátum előfordul.

K - kijelölés kulcsszavak szerint

k_1 : a kívánt kulcsszavak száma (max. 20)

k_2, k_3, \dots : a kívánt kulcsszavak (vagy jellegzetes betűcsoportjaik, lásd az alábbi megjegyzéseket)

Kijelölődnek azok a subfile-ok, amelyekben a 2-es kérdés szerint az itt megadott kulcsszavaknak legalább az egyike előfordul.

Q - kijelölés a subfile-ban előforduló kérdések szerint

k_1 : a kívánt kérdések száma (max. 20)

k_2, k_3, \dots : a kívánt kérdések sorszáma

Kijelölődnek azok a subfile-ok, amelyekben az itt megjelölt sorszámu kérdésekre vonatkozóan vannak adatok.

N - kijelölés a subfile hovatartozása szerint (a 3-as kérdés alapján)

k_1 és k_2 : a keresett név jellegzetes betűcsoportja (min. 1, max. 8 betű; lásd az alábbi megjegyzéseket)

X - kijelölés a subfile sorszáma szerint (amikor a subfile-ok neveit a SUBF utasítással listáztatjuk, a program kiírja azt, hogy a subfile az állandó file-ban hányadik helyen van)

k_1 : első sorszám

k_2 : utolsó sorszám

Kijelölődnek azok a subfile-ok, amelyek sorszáma a $/k_1, k_2/$ zárt intervallumba esik.

P - kijelölés a subfile-név 11. és 12. jegye szerint

k_1 : a subfile kulcsa (2 jel)

Kijelölődnek azok a subfile-ok, amelyek vagy korlátlan elérésűek, vagy amelyeknek a kulcsa a megadott 2 jel.

* * * * *

A megadható kritériumok száma 20. Példaként vegyük azt az esetet, amikor azoknak a subfile-oknak a neveire vagyunk kíváncsiak, amelyek (1) a Novovoronyezsi AE negyedik blokkjára vonatkoznak, (2) összeállítójuk Gubarjeva, (3) tartalmazznak kazettánkénti kiégésre vonatkozó adatokat és (4) korlátlan elérésűek vagy a kulcsuk ZA. Ebben az esetben az alábbi adatokat kell belyukasztanunk:

SUBF

ITEM B NVOR 4

ITEM N GUBAR

ITEM Q 3 69 70 71

ITEM P ZA

END

Végül néhány megjegyzést teszünk:

- Az N és K kritériumok esetében, tehát amikor a subfile-ban megadott szöveget hasonlítunk össze betűcsoportokkal, akkor az összehasonlítás csuszított módon történik. Ez azt jelenti, hogy a subfile-ban megadott szövegben előfordul-e valahol, esetleg szavak belsejében a keresett betűcsoport. Ez a keresési mód nemcsak túlságosan hosszú szövegek megadását takarítja meg a felhasználók számára, hanem bizonyos helyesírási bizonytalanságok kiküszöbölését is lehetővé teszi. (Ilyenek gyakran fellépnek cirillbetűs nevek latinbetűs átírásakor.)
- Az egymás után következő kritériumok egymással a logikai "ÉS" kapcsolatban vannak. Más szavakkal azok a subfile-ok jelölődnek ki, amelyekre mindegyik megadott kritérium egyidejűleg teljesül. Ezzel szemben áll a K és Q kritériumokon belül érvényes logikai "VAGY" kapcsolat. Ha a fenti példánkban szereplő ITEM Q kritériumot három külön kritériumra bontanánk:

ITEM Q 1 69

ITEM Q 1 70

ITEM Q 1 71

akkor azok a subfile-ok jelölődnének ki, amelyekben a 69, 70, 71 kérdések mindegyike szerepel (nem pedig valamelyike, mint a fenti példában). Ez különösen a kulcsszavas kritérium (ITEM K) esetében teszi lehetővé változatos kombinációk használatát.

- Amikor az összes subfile-okat kívánjuk kijelölni, akkor az ITEM kulcsszó helyett az ALL kulcsszót használjuk. Például:

SUBF

ALL

END

Ennek hatására megkapjuk a rendszerben lévő minden korlátlan elérésű subfile nevét.

- Implicit subfile-kijelölés csak az állandó file-ban lévő subfile-okra vonatkozhat. Ezzel szemben az explicit kijelölés mind az állandó file-ra (80 és 81), mind az ideiglenes file-ra (77) vonatkozik, mégpedig úgy, hogy a megadott subfile-nevet a program először az ideiglenes file-ban keresi,

és ha ott nem találja, akkor folytatja a keresést az állandó file-ban. Ha több azonos nevű subfile van a rendszerben, akkor mindig közülük az első jelölődik ki. Az azonos nevű subfile-ok közül a későbbiek kijelölésére az explicit kijelölést meg kell ismételni.

IV.4. ÚJ SUBFILE-OK LÉTREHOZÁSA (NEW)

Új subfile-okat a NEW utasítás segítségével hozhatunk létre. Az adatok általános formáját a IV.2. részben megmutattuk. A létrehozandó subfile neve után megadandó többi adatra vonatkozó szabályok /3/-ban találhatók részletesen. Itt csak néhány általános tudnivalót ismertetünk.

A NEW utasításhoz csak az explicit subfile-kijelölés használható. Ez a művelet természetéből következik. Egy utasítást követően legfeljebb 20 subfile-t lehet létrehozni. Ha ennél többre van szükség, akkor meg kell ismételni a NEW utasítást. Ha valamelyik subfile-ra vonatkozó adatok között hibát talált a program, akkor a hiba észlelése után következő adatkártyákat feldolgozás nélkül végigolvassa a következő subfile-névig vagy utasításig, és eközben a fel nem dolgozott kártyákat ki is nyomtatja (akár használtuk az INPT kulcsszót, akár nem). Adathibás subfile-ok nem jönnek létre. Ez a további adatokat annyiban befolyásolhatja, hogy ha később hivatkozunk erre a subfile-ra, akkor a program nem fogja megtalálni, tehát ez a dolog további hibaüzeneteket okozhat. Az egy NEW utasítás hatására sikeresen létrehozott subfile-ok neveit a program a későbbiek számára megjegyzi.

A létrehozott subfile-ok a 77-es számú ideiglenes file-be kerülnek olyan sorrendben, ahogy létrejöttek. Innen a későbbiekben listázhatók, javíthatók stb., de a program futása után automatikusan törlődnek. Az RFIT-ED program megengedi több azonos nevű subfile létrehozását. E tekintetben az RFIT adatrendszer felfogása eltér a számítástechnikában általános gyakorlattól. Ha olyan nevű adatmezőt viszünk fel a rendszerbe, amilyen ott már van, akkor ez más rendszereknél vagy a korábbi mező törlését eredményezi, vagy hibaüzenetre vezet. Nem kívánjuk megvitatni melyik felfogás a helyesebb. Véleményünk szerint mindkettőnek a maga helyén megvan a létjogosultsága. A mi felfogásunk a subfile-ok logikus, félreértéstől mentes kezelését teszi lehetővé, de a felhasználónak több szabadságot engedélyez. Ugyanakkor igaz, hogy egy esetleges figyelmetlenség könnyebben vezethet futási hibához, viszont - és ez a fontos - jobban védi a már korábban létrehozott subfile-okat, mert figyelmetlenségből nem törlődnek subfile-ok, legfeljebb két azonos nevű keletkezhet.

IV.5. MEGLÉVŐ SUBFILE-OK JAVÍTÁSA (CORR)

Korábban létrehozott subfile-ok javítása, kiegészítése, kurtítása a CORR (Correction) utasítás segítségével történik. Az adatok általános formáját a IV.2. részben megmutattuk. A subfile neve után megadandó többi adat megadásának a szabályai /3/-ban találhatók részletesen. A NEW utasításhoz hasonlóan itt is csak néhány általános tudnivalót ismertetünk.

A CORR utasításhoz csak az explicit subfile-kijelölés használható. A kijelölt subfile-nevet a program először a 77-es ideiglenes file-ban keresi, ha ott megtalálja, az első ilyen nevű subfile-t fogja javítani. Ha ott a megadott nevű subfile nincs, akkor tér át az állandó file-ban való keresésre. Ha ott sincs ilyen nevű, akkor erre vonatkozó hibaüzenet keletkezik, és a javításokra vonatkozó adatokat a program (nyomtatás kíséretében) feldolgozás nélkül végigolvassa a következő subfile-névig vagy utasításig. Ugyanez az eredménye annak, ha a program hibát fedez fel a javításra vonatkozó adatok között. A javítás menete a következő: a megtalált subfile-t a program a 74-es ideiglenes file-ba másolja át. A javított subfile először a 75-ös ideiglenes file-ban keletkezik, majd a javítás sikeres befejezése esetén onnan a javított változatot átmásolja a 77-es ideiglenes file-ba az addig ott lévő subfile-ok mögé. Az egy CORR utasítással javítható subfile-ok száma legfeljebb 20 lehet. A javított subfile-ok neveit és a javított kérdések sorszámát a későbbi felhasználás számára a program megjegyzi. Minden új CORR utasítás egy új ilyen lista megnyitását eredményezi.

Az elmondottakból látható, hogy a javított subfile neve ugyanaz lesz, mint az eredetié. Ezen a módon könnyen keletkezhetnek azonos nevű subfile-ok, mert a javítás nem eredményezi a javított subfile eredeti változatának az automatikus törlését. Ebből következik, hogy a régi változat törléséről, vagy az új változat átnevezéséről a felhasználónak kell gondoskodnia. Végül megjegyezzük, hogy a CORR utasítás használható az állandó file-ból a 77-es ideiglenes file-ba való másolásra. Ennek módja a következő:

CORR

subfile-név

NONE

Itt tehát közöljük a programmal, hogy nem adunk meg javításra vonatkozó adatokat. Egy ilyen input-adatsor hatására a 74-es és 75-ös ideiglenes file-ok közbeiktatása nélkül, eredeti alakjában jelenik meg a 77-es ideiglenes file-ban a kijelölt subfile.

IV.6. SUBFILE-OK FELÍRÁSA AZ ÁLLANDÓ FILE-BA (RCRD ÉS DMMY)

Az előzőekben láttuk, hogy az újonnan létrehozott vagy javított subfile-ok a 77-es ideiglenes file-ban keletkeznek. Ezeknek az állandó file-ba való átvitelére szolgál az RCRD (Record) utasítás. Ehhez sem subfile-kijelölést, sem egyéb adatokat nem lehet megadni. A művelet a következő három lépésben történik:

- (1) Először a subfile-ok felíródnak a 80-as és 81-es file-okba és a 60-as mágnesszalagra. Minden subfile felírása csak akkor kezdődik meg, amikor az előző felírás a 60-as mágnesszalagon is befejeződött. Minden subfile felírása után írja ki a program a felírásra vonatkozó üzenetét.

- (2) Az összes subfile-ok felírásának a befejezése után az új subfile-okat a 80-as és 81-es file-okból a program átmásolja a 61-es mágnesszalagra. Ennek befejezése után szintén kapunk egy erre vonatkozó üzenetet.
- (3) Végül a 77-es file törlődik.

Az új subfile-ok a már meglévők végére íródnak fel. Mint már utaltunk rá, a program nem vizsgálja meg vannak-e ott már olyan nevű subfile-ok, amelyeket most fel akarunk írni.

Az elmondottakból következik, hogy minden felírás feltételezi, hogy az adatrendszer nem üres, hanem ott már vannak korábban felírt subfile-ok. Ilyen körülmények között szükséges az egész rendszert valamilyen módon inicializálni. Erre szolgál a DMMY (Dummy) utasítás. Ennek hatására egy

MEASOLOLOL

nevű, egyetlen sorból álló subfile íródik fel a rendszerbe, ami lehetővé teszi, hogy saját subfile-jainkat utána felírjuk. A későbbiek során ezt a képzetes subfile-t akár ki is törölhetjük. Vigyáznunk kell azonban, hogy a DMMY utasítást csak a rendszer létrehozása előtt és csak egyszer alkalmazzuk, mert megsemmisíti az állandó file-ok korábbi tartalmát.

IV.7. SUBFILE-OK MÁSOLÁSA (COPY, REST, ADD)

Subfile-oknak a diszkek és mágnesszalagok között különböző irányokban való másolására három utasítás szolgál: COPY, REST és ADD.

A) COPY

A COPY utasítás a subfile-okat a diszkről (80 és 81) mágnesszalagra másolja. Két esetet különböztetünk meg: amikor alkalmazunk (explicit vagy implicit) subfile-kijelölést és amikor ilyent nem alkalmazunk. Az utóbbi esetben a diszken lévő összes subfile-ok átmásolódnak a 61-es mágnesszalagra.

Az első esetben a másolás egy eddig nem említett, 62-es jelű mágnesszalagra történik, és csak a kijelölt subfile-ok másolódnak át. Ha egymás után többször alkalmazzuk a COPY utasítást, akkor az első utasítás a másolást a szalag elején kezdi, de a többi utasítást az előzők végén folytatja. A COPY utasítás arra való, hogy más intézmények számára készítsünk el átadásra szánt subfile-okat, amelyeket ott az ADD utasítással fognak a saját rendszerükbe felvenni.

B) REST

A REST utasítás lehetővé teszi a mágnesszalagon (60) tárolt subfile-oknak a diszkre való másolását. Ennek eredeti rendeltetése az, hogy a diszken őrzött információt a mágnesszalagok alapján visszaállítsuk. (Erre utal az utasítás neve is: restore.) A visszaállítás a 60-as mágnesszalagról történik. A visszaállítás befejezése után nem készül automatikusan másolat a 61-es mágnesszalagra, mert normális körülmények között kell lennie két azonos tartal-

mu mágnesszalagnak. Ha nem ez a helyzet, akkor külön alkalmazni kell a COPY utasítást, mégpedig subfile-kijelölés nélkül, hogy a másolat a 61-es mágnesszalagon keletkezzen. (A COPY utasításnak ez a kettős használata ebből a célból született.)

A REST utasítás felfogható egy egyszerű mágnesszalag-diszk másolási műveletnek is. Ilyenkor rendszerint alkalmazunk (implicit vagy explicit) subfile-kijelölést, amikor is csak a kijelölt subfile-ok másolására kerül sor. Ha a REST utasítást többször egymás után alkalmazzuk, akkor először a program diszket üresnek tekinti, de a későbbi utasítások hatására a másolás az előzők folytatása lesz. A legvégén természetesen célszerű (subfile-kijelölés nélküli) COPY utasítás kiadása, hogy két azonos mágnesszalagunk legyen.

C) ADD

Az ADD utasítás hatására a 62-es mágnesszalagon található subfile-ok az állandó file-okba másolódnak folytatva az ott már meglévő subfile-ok sorát. A másolás menete a következő: a másolandó subfile-ok először a 62-es mágnesszalagról a 60-as mágnesszalagra másolódnak, majd ennek végeztével ugyanez másolódik párhuzamosan a diszkre és a 61-es mágnesszalagra. A másolt subfile-ok felsorolását a program ebben a második lépésben nyomtatja ki, tehát biztosak lehetünk abban, hogy a kinyomtatott subfile-ok a rendszerbe szabályosan felkerülnek. (Ilyen aggályok akkor szoktak felmerülni, amikor valamilyen futási hiba, pl. időtullépés miatt a másolás félbeszakadt.) Az ADD utasításhoz adhatunk meg explicit vagy implicit subfile-kijelölést, de ez el is maradhat, mert ekkor minden subfile átmásolódik.

Másolás közben a program megnézi, "ismeri-e" azokat a subfile-neveket, amelyek a 62-es mágnesszalagon vannak. Ha ilyen nem szabványos nevű subfile-ok vannak a szalagon, ezeket a program nem másolja át.

IV.8. SUBFILE-OK TÖRLÉSE (DEL, DEL7)

Subfile-okat törölhetünk akár az állandó file-okból (DEL utasítás), akár a 77-es ideiglenes file-ból (DEL7). Mindkettőhöz kell használnunk subfile-kijelölést, és ez lehet akár explicit, akár implicit. A DEL művelet az egyetlen, amely helyrehozhatatlan módon érinti az állandó file-okat, mert a bennük lévő információ törlésével jár együtt. Ezért ezt nagy körültekintéssel kell végezni, és, ha csak lehet, ritkán alkalmazzuk: az állandó file-okba csak kifogástalan subfile-ok kerüljenek fel. Ha mégis sor kerül a DEL utasítás alkalmazására, akkor egyszerre lehetőleg több subfile-t töröljünk. A szükségtelen törlések elkerülésének a legjobb módja a DEL7 és RNM7 (l. alább) utasítások kellő időben való alkalmazása.

A) DEL

A DEL (Delete) utasítás hatására a subfile-ok az állandó file-okból az alábbi lépésekben törölődnek. Először a ki nem jelölt subfile-ok átmásolódnak a 60-as mágnesszalagra (megsemmisítve annak korábbi tartalmát). Ezután a program végrehajt egy REST utasítást. A 61-es szalagra való másolás nem történik meg automatikusan, tehát erről külön kell gondoskodni (mint a REST esetében is). Ebből is látható, hogy a DEL utasítás nagyon hosszadalmas műveletet eredményezhet, ha a rendszerben már sok subfile van.

B) DEL7

A DEL7 (Delete 77) utasítás hatására a kijelölt subfile-ok kitörölődnek az ideiglenes 77-es file-ból. Ez a művelet sokkal egyszerűbb, kevesebb időt is igényel, mint a DEL, ezért a felesleges subfile-okat a RCRD utasítás előtt DEL7-tel törölni célszerű.

IV.9. SUBFILE-OK TARTALMÁNAK A LISTÁZÁSA (LIST)

A subfile-ok tartalmát a LIST utasítás segítségével listázhatjuk. Az utasítás használata nagyon egyszerű: az utasítással együtt megadott (explicit vagy implicit) subfile-kijelölés szerinti subfile-ok tartalmát egy áttekinthető formában listázva kapjuk meg. Felhívjuk a figyelmet arra, hogy ez a forma különbözik mind az input adatok formájától, mind attól, hogy az adatok a rendszerben tárolódnak.

A LIST utasításnak vannak szűkitett változatai, amelyek nem a teljes subfile-tartalom listázását eredményezik, hanem csak egy részét. Az említett változatok a következők:

TLST: szöveges kérdések (1-50) listázása (list of text)

FLST: számszerű kérdések (51-99) listázása (list of figures)

CLST: azoknak a kérdéseknek a listázása, amelyeket korábban javítottunk (emlékeztetünk arra, hogy javításkor a program subfile-onként megjegyzi a javított kérdések sorszámait; correction list)

PLST: megadott sorszámú kérdések listázása (megadandó: a kérdések száma, és a kérdések sorszámai; partial list).

A NEW utasítás végrehajtása után automatikusan végrehajtódik egy LIST utasítás, a CORR után pedig egy CLST utasítás. A listázó utasítások ugyanazokra az utasításokra vonatkoznak, mint a NEW ill. CORR. Ha ezekre a listákra nincs szükség, ezt le lehet tiltani az NLST (no list) utasítással, amelyet a NEW vagy CORR utasítással egy mondatban kell megadni. (Az utasításmondatókat illetően lásd IV.12.) Az NLST utasítás tulajdonképpen egy üres utasítás, amelynek semmi hatása nincs, bármikor alkalmazhatjuk.

IV.10. A KATALÓGUSRA VONATKOZÓ UTASÍTÁSOK

Vannak utasítások, amelyek a subfile-ok tartalmát nem érintik, csak a subfile-ok katalógusára vonatkoznak. Ezek a következők:

A) SUBF

A SUBF (subfiles) utasítás hatására kilistázódnak a kijelölt subfile-ok katalógusában tárolt adatok (lásd III.1). Egyaránt alkalmazhatunk implicit és explicit kijelölést. Erről a listáról olvashatjuk le az ITEM X kijelöléshez szükséges sorszámokat is.

B) SFMT

AZ SFMT (subfiles on magnetic tapes) utasítás hatása ugyanaz mint a SUBF utasításé, de a katalógus-adatokat nem a 80-as diszk-file-ből, hanem a 60-as mágnesszalagról vesszük.

C) COMP

A COMP (compare) utasítás segítségével összehasonlíthatjuk, azonos-e a diszken tárolt file tartalma a 60-as és 61-es mágnesszalagokon tárolt file-okkal. Az összehasonlítás nem terjed ki a subfile-ok belső tartalmára, hanem csak a katalógusokéra (tehát azokra a mennyiségekre, amelyeket a SUBF utasítás hatására kilistáztatunk). Ha az összehasonlítás azt mutatja, hogy a file-ok nem azonosak, akkor a további utasítások végrehajtására már nem kerül sor. A COMP utasítást tehát akkor célszerű használni, ha olyan műveleteket akarunk végezni, amelyekhez szükséges a file-ok azonossága (pl. ADD). A COMP utasításhoz természetesen nem szabad subfile-kijelölést megadni.

D) RENM

Az állandó file-okban tárolt subfile-ok nevének a megváltoztatására szolgál a RENM (rename) utasítás. Ehhez csak az explicit subfile-kijelölés alkalmazható. Először megadjuk a subfile régi nevét, majd (külön kártyán) az új nevet. Az egy utasítással átkeresztelhető subfile-ok száma legfeljebb 10 lehet. A program az új neveket megjegyzi egy későbbi felhasználás számára.

Az RENM utasítás hatására a subfile az új néven, javított katalógus-adatokkal átmásolódik a 77-es ideiglenes file-ba az ott lévők mögé, de az állandó file-okba való visszamásolás és a régi subfile törlése nem történik meg automatikusan. Erről külön RCRD és DEL utasításokkal kell gondoskodni. Ha több azonos nevű subfile van az állandó file-okban, akkor közülük az első veszi a program. Ha a többit is át akarjuk keresztelni, akkor a közös régi nevet annyiszor kell megadni ahányadiknak a nevét akarjuk megváltoztatni. Látható ebből, hogy az RENM utasítás használata a lehetőség szerint ugyanugy kerülendő, mind a DEL utasításé: törekedni kell arra, hogy minden subfile a

végleges néven kerüljön fel az állandó file-okba. Ennek jó eszköze az RNM7 utasítás (lásd alább). Megjegyezzük még, hogy egy subfile titkosítása, vagy a titkosítás feloldása is az RENM utasítással történik.

E) RNM7

A 77-es ideiglenes file-ban lévő subfile-ok nevének a megváltoztatására szolgál az RNM7 (rename 77) utasítás. A régi és új subfile-nevek megadása ugyanugy történik, mint RENM esetében. Az RNM7 utasítás a 77-es file keretei között marad, emiatt használata egyszerűbb, mint az RENM utasításé. A fő különbség abban van, hogy az átkeresztelt subfile-ok a helyükön maradnak (hiszen ez nem jár mágnesszalagok tartalmának a felülírásával). Az adatok megadásánál figyelembe kell venni, hogy minden subfile-átnevezés a megelőző átnevezések eredményéből indul ki, beleértve az átnevezendő subfile-ok keresését is.

IV.11. SPECIÁLIS UTASÍTÁSOK (DATE, TITL, END)

Befejezésül ismertetünk három speciális utasítást, amelyek az RFIT-ED program használatával, de nem a subfile-okkal kapcsolatosak.

A) Cim és dátum

A program által készített minden line printer oldalon szerepel valamilyen dátum és cím. Ez az adott programváltozattól függően valamilyen szöveg. Ha súlyt helyezünk arra, hogy az általunk készített eredményen a saját szövegünk szerepeljen, akkor a TITL (title) utasítás segítségével megadhatjuk ezt az alábbi egyszerű módon:

TITL

általunk kívánt cím

Ettől kezdve minden oldalon az itt megadott cím fog szerepelni. Ha vissza akarunk térni a szokásos címhez, akkor elég a TITL utasítást önmagában megadni, és attól kezdve ismét a szokásos cím fog megjelenni minden oldalon. Ez a KFKI változatnál: "Manipulations with the stored data".

Hasonlóan, a futás dátumát is megadhatjuk:

DATE

új dátum

Ettől kezdve minden oldalra az új dátum fog kerülni. Ha ezt nem tesszük, akkor a szokásos dátumot fogjuk minden oldalon találni: 23/02/72. Erre úgy tudunk visszatérni, hogy a DATE utasítást önmagában adjuk meg.

B) END

Fentebb többször volt szó arról, hogy az input adatok olvasása utasítástól utasításig történik. Annak érdekében, hogy az utolsó utasításhoz tartozó

adatok feldolgozása is rendben megtörténhessen, az adatok végére célszerű az END utasítást tenni, aminek a hatására a program futása befejeződik. Ennek elhagyása rendszerint hibát eredményez.

IV.12. UTASÍTÁS-MONDATOK

A különböző utasításokat általában külön kártyán, a kártya bal szélén szoktuk megadni. Van azonban arra is lehetőség, hogy egyes, összetartozó utasításokat egy kártyán, egymás után lyukasztva adjunk meg. Ezeket nevezzük utasítás-mondatoknak. Egy mondat legfeljebb 5 utasításból állhat. Mondatok megadása akkor célszerű, ha több utasítás van, amelyekhez ugyanaz a subfile-kijelölés tartozik. Nézzük egy példát. Tegyük fel, hogy néhány subfile-t (amelyek mind a 77-es file-ban vannak) ki akarunk javítani, az eredeti változatokat törölni, majd a javítottakat listázni és az állandó file-okba felírni akarjuk. Nyilvánvaló, hogy mindegyik művelethez azonos subfile-kijelölés tartozik. Ezért a következő mondat használata biztosítja legegyszerűbben a kívánt műveletek végrehajtását:

CORR DEL7 LIST RCRD

A CORR hatására mindegyik subfile-ből két példány keletkezik: a 77-es file elején helyezkednek az eredeti változatok, a végén a javítottak, sőt a CORR végrehajtása során a program meg is jegyzi a javított subfile-ok neveit. A DEL7 utasítás kitörli a megjegyzett nevű subfile-okat. Tekintve, hogy előbb az eredeti változatokat találja meg, végeredményben a javított változatok fognak megmaradni. Ezután a LIST a javított változatokat listázza, majd az RCRD utasítás ezeket felírja az állandó file-okba. Vegyük észre, hogy ez akkor is jól működik, amikor egyes subfile-ok javítása sikertelen volt: ilyenkor ezek nevei nem kerülnek fel arra a listára, amelynek alapján a DEL7 töröl, tehát a sikertelenül javított subfile-ok érintetlenül maradnak.

Ennek a példának az alapján hasonló mondatok könnyen szerkeszthetők, ha jól értjük az egyes utasítások működését.

IRODALOM

- [1] Gadó J.: A VVER-440 típusu atomerőművi reaktorok számítására szolgáló BIPR program ismertetése. KFKI-1978-72 report, 1978
- [2] Ф.Я.Овчинников: Эксплуатационные режимы водо-водяных энергетических ядерных реакторов, Атомиздат 1977
- [3] З.Сатмари: Программа RFIT-ED для обслуживания системы хранения нейтронно-физических данных эксплуатации АЭС типа ВВЭР, KFKI-ZR-6/415/1981
- [4] Z. Sztarmáry, User's manual of program RFIT, előkészületben. KFKI report-ként



Kiadja a Központi Fizikai Kutató Intézet
Felelős kiadó: Gyimesi Zoltán
Szakmai lektor: Valkó János
Gépelte: Beron Péterné
Példányszám: 55 Törzsszám: 82-469
Készült a KFKI sokszorosító üzemében
Felelős vezető: Nagy Károly
Budapest, 1982. szeptember hó

63.284